displaying the presentation document on the output device coupled to the client system with the presentation document containing content from the defined content document within the defined area within the presentation document.

4. A computer readable medium having a template stored thereon said template defining a layout for content to occupy, an engine capable of placing content within the defined layout, the template comprising:

a tag defining a container area; **[and]**

wherein **[the]** an HTML engine upon recognizing the tag places a selected content within the defined container area.

5. A method for customizing textual mark-up documents, a textual mark-up layout template having a tag, said tag defining a layout area, the method comprising:

defining a textual mark-up **[document]** content source; and

flowing the textual mark-up **[language]** content source into the layout area defined by the tag in the textual mark-up layout template **[defined by the tag]**.

6. A method for customizing textual mark-up documents, a textual mark-up content document, a textual mark-up customizing document, a textual mark-up parsing engine, the method comprising:

parsing in a tag defining a container area and attributes from the textual mark-up customizing document;

instantiating a container **[component]** object in response to parsing in the tag defining the container area and attributes;

placing the container area and attributes into the container **[component]** object; and

flowing the textual mark-up content document into the area defined by the container area and attributes placed in the container **[component]** object.**[;]**

[breaking the flow of the textual mark-up content document into the area defined by the container area and attributes placed in the container component object upon an indication that the area defined by the container area and attributes is full; and

saving the position in the textual mark-up content document where the flow stopped.]

8. The method of claim 7, further comprising:

flowing a textual mark-up language element into more than one of the areas defined **[in]** by the series of tags in the textual mark-up customizing document; and

maintaining a chain of records of **[the]** textual mark-up language elements that are flowed into more than one of the areas defined **[in]** by the series of tags in the textual mark-up customizing document.

13. A method for customizing textual mark-up documents on a computer network using a textual mark-up content source **[document]** containing textual mark-up language elements, a textual mark-up customizing document containing a series of tags and a textual mark-up parsing engine, the method comprising:

accessing locally or remotely the textual mark-up customizing document containing a series of tags, the tags defining a series of containers for placing a textual mark-up content source **[document]**; and

accessing locally or remotely the textual mark-up content source **[document]** to be flowed into the series of defined containers.

Please add the following claims:

-- 14. A method for recursively customizing a textual mark-up document, the method comprising:

parsing in a textual mark-up behavioral tag that identifies a first-generation area to accept textual mark-up source content and a first-generation textual mark-up source content to occupy the first-generation area; and

within the first-generation textual mark-up source content a second-generation textual mark-up behavioral tag is encountered that identifies a second-generation sub-area within the first-generation area to accept textual mark-up source content and a second-generation textual mark-up source content to occupy the second-generation sub-area.

15. The method of claim 6 further comprising:

breaking the flow of the textual mark-up content document into the area defined by the container area and attributes placed in the container object upon an indication that the area defined by the container area and attributes is full; and

saving the position in the textual mark-up content document where the flow stopped.

16. A method of integrating textual mark-up content sources having the bottomless-page model in accordance with a custom layout on a computer having an output device, an operating system operating under the control of a textual mark-up engine, a textual mark-up template document, the method comprising:

parsing in a tag in the template document, the tag defining a textual mark-up content source and an output area within the output to accept the textual mark-up content source;

flowing the textual mark-up content source into the defined output area;

parsing in a tag in the template document, the tag defining a second textual mark-up content source and a second output area within the output to accept the second textual mark-up content source; and

flowing the second textual mark-up content source into the defined second output area.

17. The method of claim 11, wherein for each separate layout flow object instantiated, if that layout flow object remains in scope upon an indication that the area defined by a tag is filled up, then each such separate layout object creates a break object that identifies the state of the layout of the textual mark-up language element it represents.

18. The method of claim 17, wherein all break objects created for layout flow objects that remain in scope upon an indication that the area defined by a tag is filled up, are ordered in a break record and made accessible based on a distinct context identifier assigned to each area defined by a tag.

19. The method of claim 13, where the textual mark-up content source containing textual mark-up language elements contains both in-flow content elements and positioned content elements, and a record maintains a running total of the sum of the distances occupied by the series of containers defined for a textual mark-up content source, the method comprising:

determining after parsing in a tag whether an explicit offset request for placement of a positioned element requests placement at an explicit offset that exists within the container area defined by the tag based on where the dimensions of the container exist within the running total of the sum of the distances occupied by the series of containers.

20. A method for customizing textual mark-up documents on a computer network using a textual mark-up content source, a textual mark-up customizing document containing a series of tags and a textual mark-up parsing engine, the method comprising:

parsing in a first tag in the textual mark-up customizing document, the first tag defining a container area and a first textual mark-up content source to be flowed into the defined container area, and the first tag further defining a second tag expected to be found in the customizing document, the second tag to receive an overflow of the first textual mark-up content source if that overflow would not fit within the container area defined by the first tag;

placing in a record a tuple, the tuple associating a unique first tag identifier with a unique second tag identifier;

21. The method of claim 20, further comprising:

parsing in a second tag in the textual mark-up customizing document, the second tag defining a container area; and

determining from the tuple in the record based on associations of unique identifiers, that the overflow of the first textual mark-up content defined in the first tag should be placed in the container area defined in the second tag.

22. The method of claim 21 wherein textual mark-up content defined in the customizing document is placed into areas in between the container areas defined in the first and second tags.

23. The method of claim 21, further comprising:

before parsing in the second tag, parsing a third tag in the textual mark-up customizing document, the third tag defining a container area and a second textual mark-up content source to be flowed into the container area defined by the third tag, and the third tag further defining a fourth tag expected to be found in the customizing document, the fourth tag to receive an overflow of the second textual mark-up content source if that overflow would not fit within the container area defined by the third tag;

determining from the record that the third tag is not related to the tuple associating the unique first tag identifier with the unique second tag identifier; and

placing in the record a tuple, the tuple associating a unique third tag identifier with a unique fourth tag identifier.

24. The method of claim 20, further comprising:

parsing in an arbıtrary tag and determining from the record whether the arbitrary tag is expected to receive textual mark-up content source overflow from a previous container area defined in a previous arbitrary tag based on whether the record associates the arbitrary tag unique identifier with the previous arbitrary tag unique identifier. --

Respectfully submitted,

KLARQUIST SPARKMAN CAMPBELL
LEIGH & WHINSTON, LLP

By Stephen A Wight
Stephen A. Wight
Registration No. 37,759

One World Trade Center, Suite 1600
121 S.W. Salmon Street
Portland, Oregon 97204
Telephone: (503) 226-7391
Facsimile: (503) 228-9446